

Ejercicio 4:

```

4) a) function suma Vector 64 (vector, longitud, suma)
    iterador = 0x0000
    suma[4]
    if (longitud == 0) < 0x0004
        while (iterador < 0x0004)
            suma[iterador] = 0x0000
            iterador = iterador + 1
    else
        while (iterador < 0x0004)
            suma[iterador] = vector[iterador]
            iterador = iterador + 1

```

```

    while (iterador < longitud * 4)
        sumar64 (suma, vector[iterador ... iterador + 4], suma)
        iterador = iterador + 4
    return

```

```

b) main:    CMP R0, 0x0000
            JNE longitud1
            MOV [R3], 0x0000
            MOV [R3+0x0001], 0x0000
            MOV [R3+0x0002], 0x0000
            MOV [R3+0x0003], 0x0000
            JMP fin

longitud1:  MOV [R3], [R1]
            MOV [R3+0x0001], [R1+0x0001]
            MOV [R3+0x0002], [R1+0x0002]
            MOV [R3+0x0003], [R1+0x0003]

```

R0 → longitud del vector
 R1 → puntero al vector
 R3 → puntero a la suma total
 R4 → comparador con la longitud

```

            CMP R0, 0x0001
            JE fin

ciclo Suma: MOV R4, R0+0x0004
            ADD R1, 0x0004
            ADD R3, 0x0004
            MOV R2, R3
            MOV R0, R3
            SUB R4, 0x0001

ciclo Suma: CALL Sumar64
            SUB R4, 0x0001
            CMP R4, 0x0000
            JNE ciclo Suma

fin:        RET R0, R2

```